CHAPTER 1

INTRODUCTION

Over the years our world has modernized more and more with technology advancing on a whole new level. With upcoming years we are observing changes in society due to the technology changing around us. So it is safe to assume that the great growling engine of technology will keep moving forward. But the real question is “Does this technology exist for our benefit?” i.e. Is it really benefitting us or it is just a ‘Make Believe’ situation created by those ruling technology to delude us into believing that it is made for us. Well it can be either ways but we should not forget one thing that it is not what technology does to us, it is what we did with technology.

So it’s time to get smart with the technology, choose wisely and use it in a way that benefits, both us and those around us. Working for developing new innovations is one thing and doing it for the sole benefit of society is other. The project that we intend to build is for the benefit of our society which becomes our reason to take up this project. The technology we create is not meant to impress someone, but for the experience that we gain with it is everything. By now you must be wondering, what is that we are aiming at? Before that let me tell you an interesting fact about our government. The transport budget gets 20-25 % of annual budget i.e. around INR 38,000 million/-. But is that money reaching the people? The answer is very clear it is not. So our efforts are focused on making Bus transportation more comfortable for those availing it.

1.1 Problem introduction

Our project deals with daily problems of bus travellers i.e. seat availability, time of the bus and the bus route. The main focus of the application is to handle lack of information about the vacant seats in arriving buses.

Our aim is to make the lives of bus passengers easier by providing them a means to have the idea of number of vacant seats in the arriving buses as well as expected time of arrival and the bus route that can prevent them from getting into unnecessary trouble.

1.1.1 Problem Motivation

The passenger may enter a crowded bus even if the next bus towards his destination that would arrive in 5 to 7 min is empty. This lack of information causes a lot of trouble for daily passengers who go to office or even students who use bus as a daily transport.

1.1.2 Project Objective

To successfully create a prototype of a system that implements "WITTY BUS TRACKER SYSTEM" and gives the User an idea about the seat availability in the arriving buses, time of arrival and route of the bus.

1.1.3 Scope of the Project

It can be used by government transportation organisations to provide better services to the passengers so that they can have a hassle free experience.

The data provided by our application can also be used to decide the most appropriate time intervals between the departures of two consecutive buses of the same route**.**

1.2 Related Previous Works

There are many applications available in the Google play store which tells about the route and the timing of the buses but there is no such application till now which tells the seat status in the government buses. Some of the available applications are:

1.2.1 MTC bus route

It gives details of Chennai Public Transportation buses including bus timings and route.

1.2.2 Delhi DTC bus timings and route

It gives details of Delhi Public Transportation buses including bus timings and route.

1.3 Organization of the Report

The report encloses chapter wise description of our work. The first chapter includes the basic objective of what we want to achieve through our project, how could it help everyone and why did we choose it. Second chapter contains software requirement specification i.e. roadmap to how we will achieve our goals. Chapter three includes the whole design of the system through various diagrams which explains all the different aspects of designing process. Chapter four is the summation of what we have achieved including the test cases and output. The report ends with the conclusion of what are final results and features.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

This chapter is the step by step evaluation of the product that would decide the path to be followed throughout the whole project. Successful implementation of these steps will lead to better product that will satisfy the user requirements.

2.1 Product Perspective

This product is independent and totally self-contained.

2.1.1 System Interfaces

The system's modules interact within effectively. The user module interacts with the user and the server, while the conductor module interacts with the conductor and the server. The effective interaction leads to successful execution of the whole system.

2.1.2 Interfaces

The system comprises of two types of users:

- Conductor: will be provided a GUI to enter all the details into the ticketing machine.
- App User: will use an android app to check various details related to the buses. All this will be done via a Graphical User Interface of the android phone.

2.1.3 Hardware Interfaces

- Ticketing machine: The program for entering ticket details must be run on a normal ticketing machine with Bluetooth support.
- Mobile device embedded in the bus: the mobile should have Bluetooth support, a GPS system and all time internet connectivity.
- User device: Android phone with internet access.

2.1.4 Software Interfaces

- MySQL: The system must use MySQL as its database component. The server will store the bus related details in this database.

2.1.5 Communications Interfaces

The system uses following protocols for short range as well as long range communication:

- WIFI (IEEE 802.11 standards)
- Wireless Application Protocol (WAP)

2.1.6 Memory Constraints

The minimum memory required is 512 MB RAM that comes in almost all handsets of Android operating system. Other than that the app takes up only 9.24 MB space.

2.1.7 Operations

Under normal conditions, the following operations will be performed in the system:

- Entry of ticket details by the conductor
- Transfer of details by Bluetooth device to mobile embedded in the bus.
- Local processing of data by the mobile
- Transfer of information from mobile to server
- Updation of database
- Query by the app user
- Response from server to user.

If at any point of time, there is some problem while communicating from one device to another, an error message will be displayed to the user.

Since the data is being processed locally (inside the buses), when the ideal environmental conditions are met, the system will again work perfectly.

2.1.8 Site Adaptation Requirements

The app user only needs an android device to get access to the system. On the other hand, a mobile with GPS, Bluetooth support and internet access should be embedded in all the buses and also, the ticketing machine should be provided with Bluetooth support.

2.2 Product Functions

The product will provide the following features:

- Tell the current seat availability status in any local bus..
- Provide the bus route information.
- Estimate the time of arrival at a particular stop.

2.3 User Characteristics

- App user: Any person with the basic knowledge of using an android smartphone can use the product.
- Conductor: Bus conductor with knowledge of using a normal ticketing machine can use our product easily.

2.4 Constraints

- Operating system in mobile device must be android.
- Ticketing machine must support Bluetooth
- Mobile embedded in the bus must have GPS, Bluetooth support and internet access.
- Mobile embedded in the bus must be installed at a secure place where unauthorized access is not allowed.

2.5 Assumptions and Dependencies

The following must be assumed for ensuring perfect working of the product:

- The ticketing machine supports Bluetooth which is switched on all the time.
- Mobile device embedded in the bus has all time internet access.

2.6 Use case

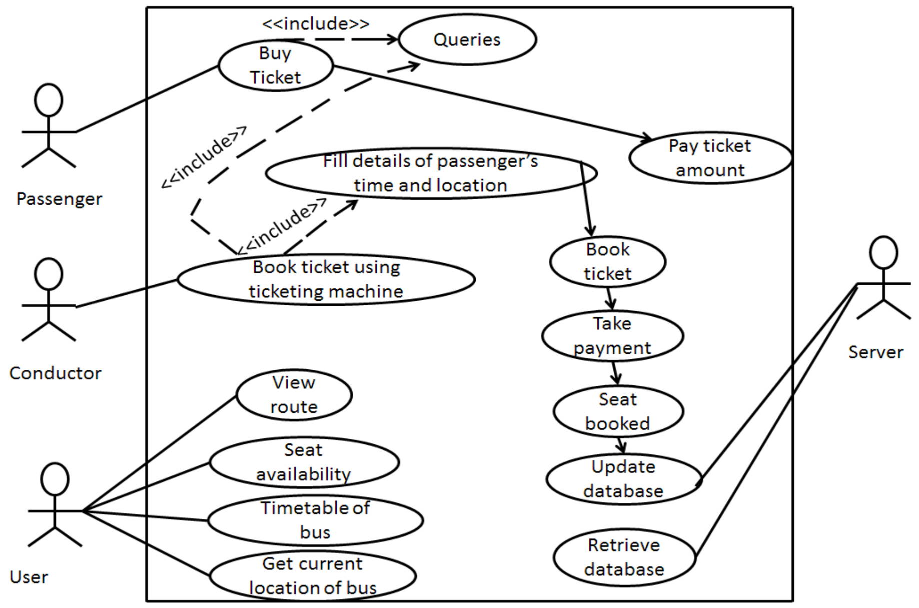


Figure 2.1: Use case Model

Table No.2.1 Details of Use Case

Use Case Name	Witty bus tracker
Actors	Passenger, conductor, User, Server
Description	User can get the knowledge of current available seats, location and route
Precondition	User must have an android- mobile
Basic flow	The user chooses the bus and gets the current seats
Alternate flow	If the requests load increase user may get incorrect information
Post condition	The server must store the entered details

As shown in Figure 2.1 (Use case model) and table 2.1 passengers, conductor, User and server are the four actors who perform certain described functions like passenger buys a ticket and conductor makes an entry of it which id updated in the database as soon as update service sends the new updates.

2.7 Sequence diagrams

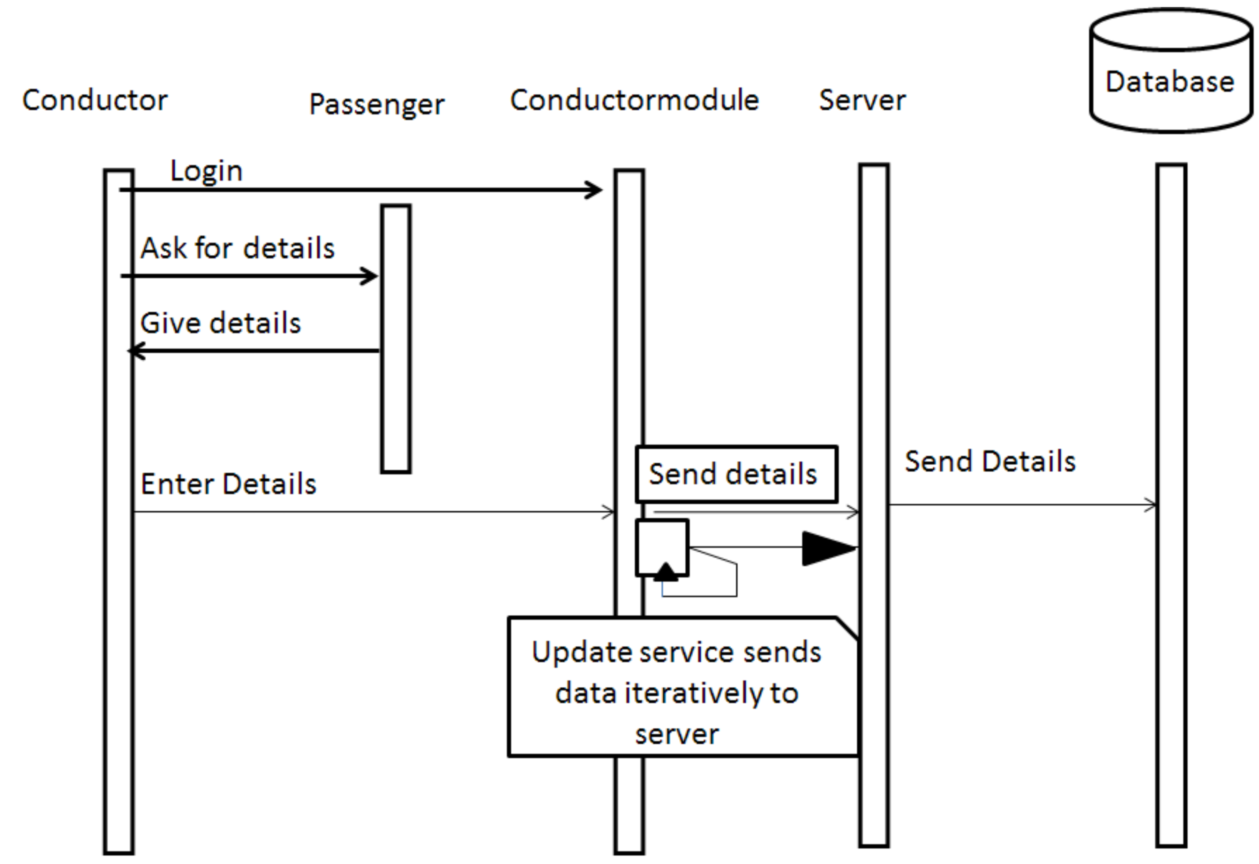


Fig 2.2: Conductor module

In Fig 2.2 the conductor logs into the conductor module then enters the details of user i.e. the number of passengers, source and destination. The details are send to the server periodically which has been shown by loop in the diagram which are then stored in the database. The update service is a background process that does not needs to be triggered by the conductor. That service is responsible for sending all the details.

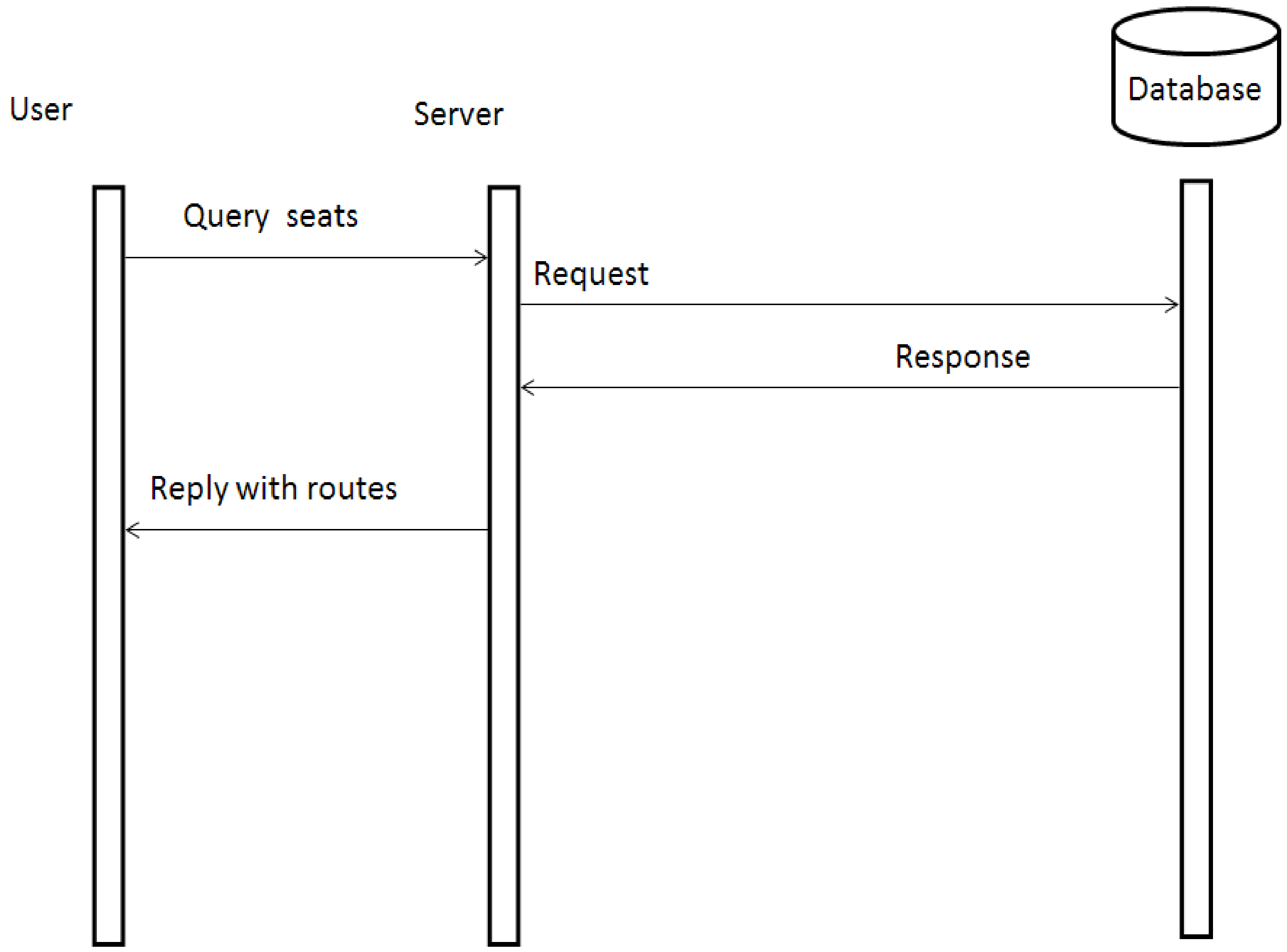


Fig 2.3: User Module

In Fig 2.3 i.e. the user end, user asks queries about route or vacant seat by entering details like route Id or source and destination. These details will be requested to server. The server replies as it is getting constant updates from the conductor module through the background update service. The reply is then relayed to the user. User can ask as many queries as it wants.

CHAPTER 3

SYSTEM DESIGN

This chapter deals with the designing phase of the project through describing various diagrams like architecture diagram, DFD, ER diagram etc. to get a better view of what we are supposed to build and how it will work.

3.1. Architecture diagrams

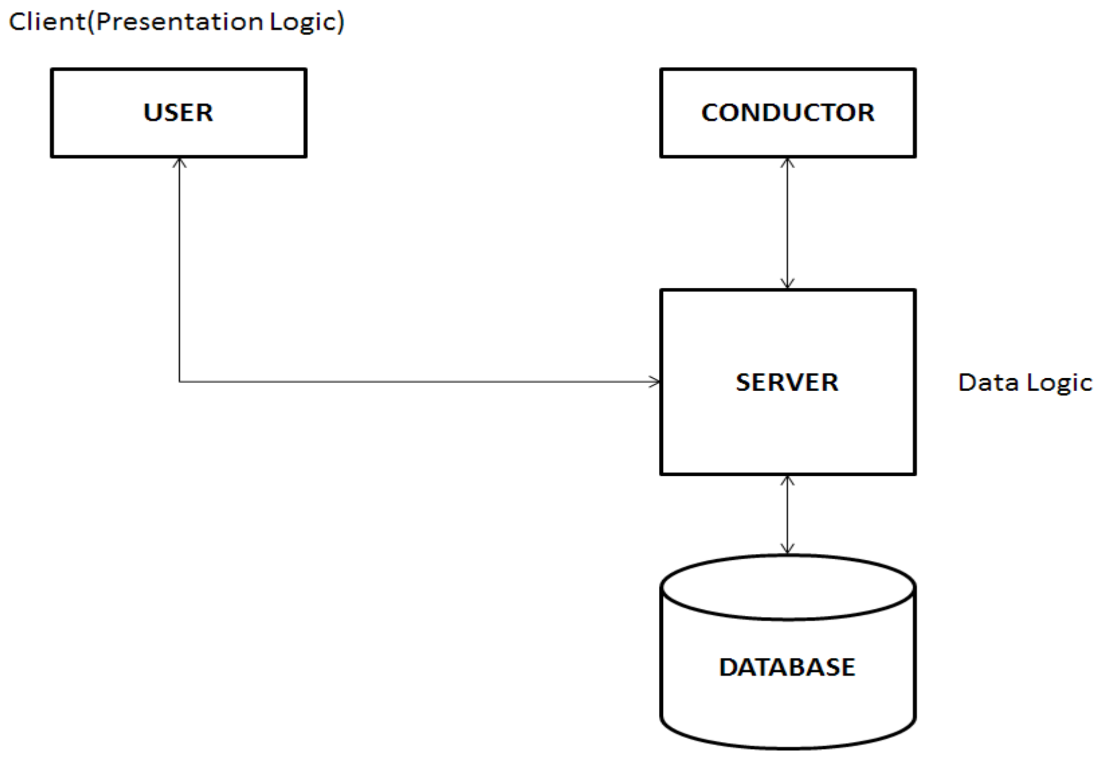


Fig 3.1: architecture of system

In Fig 3.1 the basic architecture of our project is shown. It contains User, Conductor, Server and Database. The user interacts with the server by sending queries. The queries include the number of vacant seats, route of the bus etc. Conductor is responsible for entering the details in conductor module. A background update service is responsible for sending updates of all the details to the server which are then stored in the database.

3.2.Class diagrams

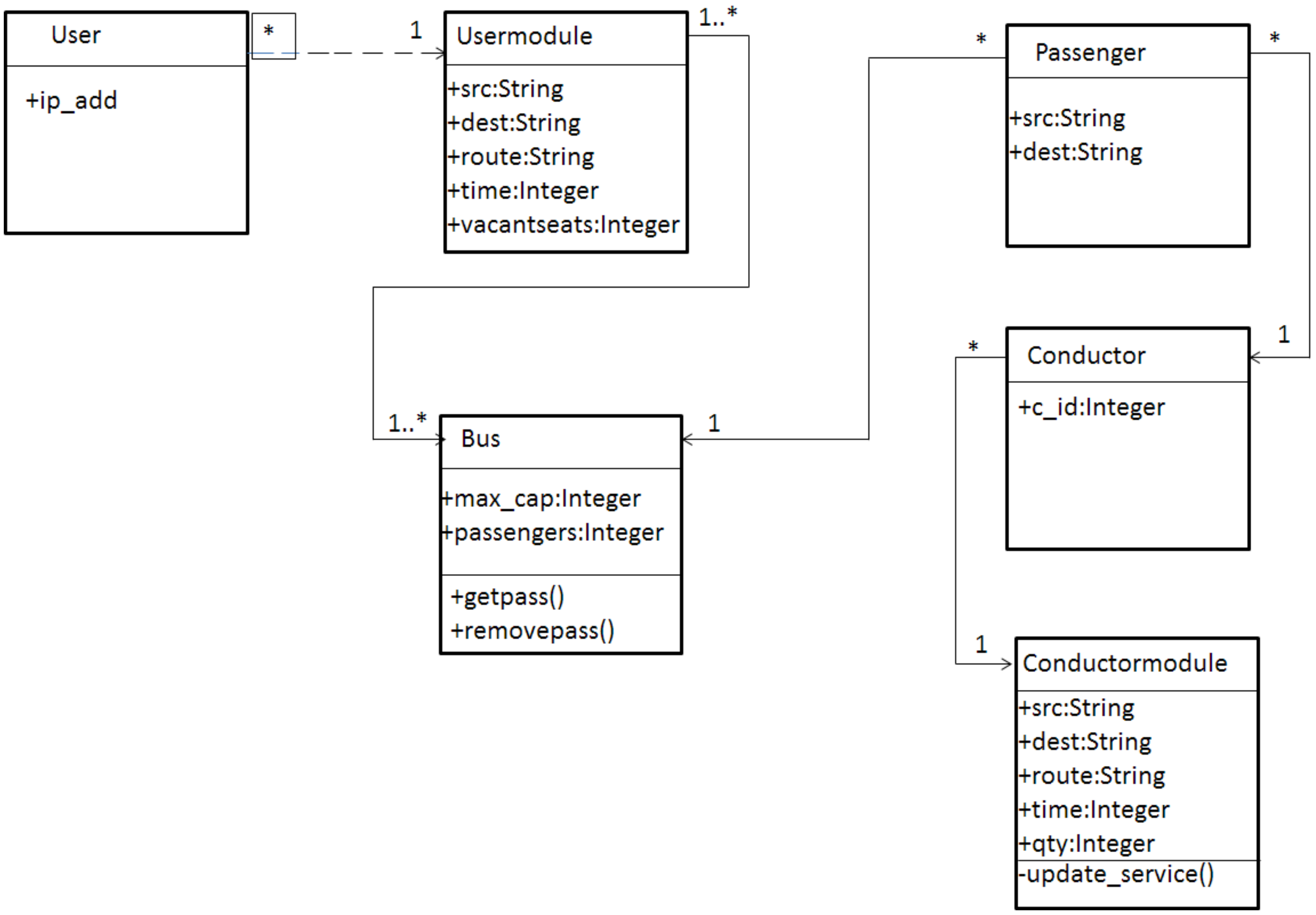


Fig 3.2: Class diagram

In diagram 3.2 the modules involved in the whole system are represented as separate classes which have their own public and private members. The further relationships have been shown through the use of one to many, many to one relations. User module has members like source, destination, route as string and time, vacant seats as integer. Passenger can get on the bus and get a ticket from the conductor who has a conductor Id and will make all the entries as per conductor module. Conductor module has an update service that keeps on going in the background in short intervals.

3.3.Data Flow Diagram

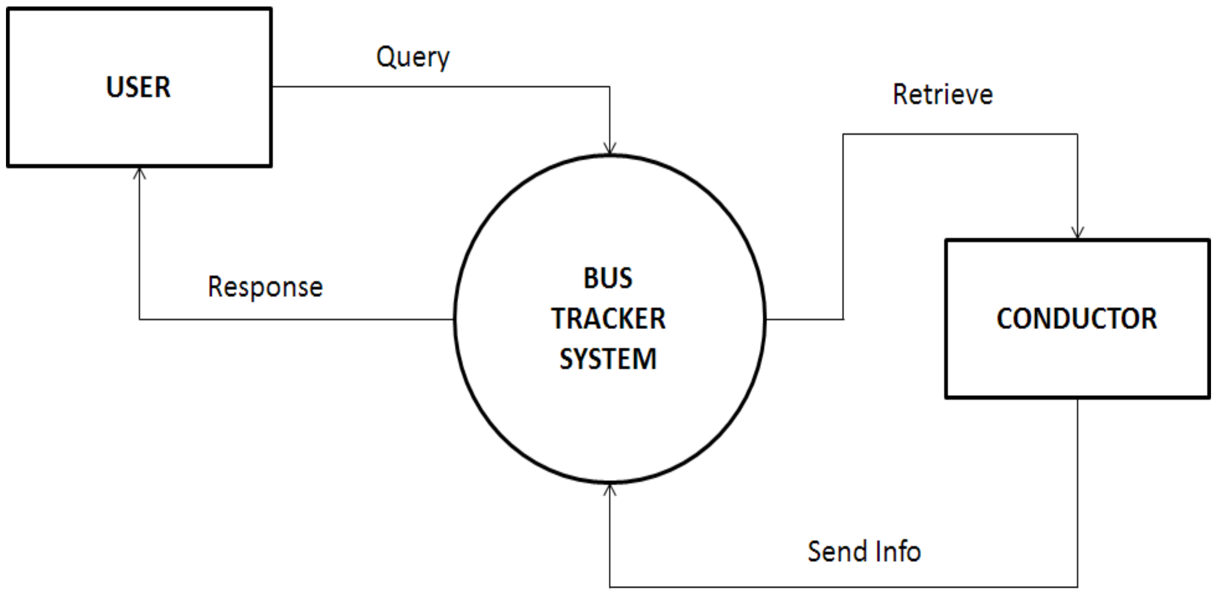


Fig 3.3: DFD LEVEL 0

Fig 3.3 represents a basic level 0 data flow diagram which represents the prototype that we are working on at its root level. The bus tracker systems interacts with both the ends i.e. user and conductor. User can send a query in user module and get response while conductor has to enter the information in the conductor module. There is an update service which keeps the system updated of the recent changes that occur regarding seats, location and time. This service runs in background.

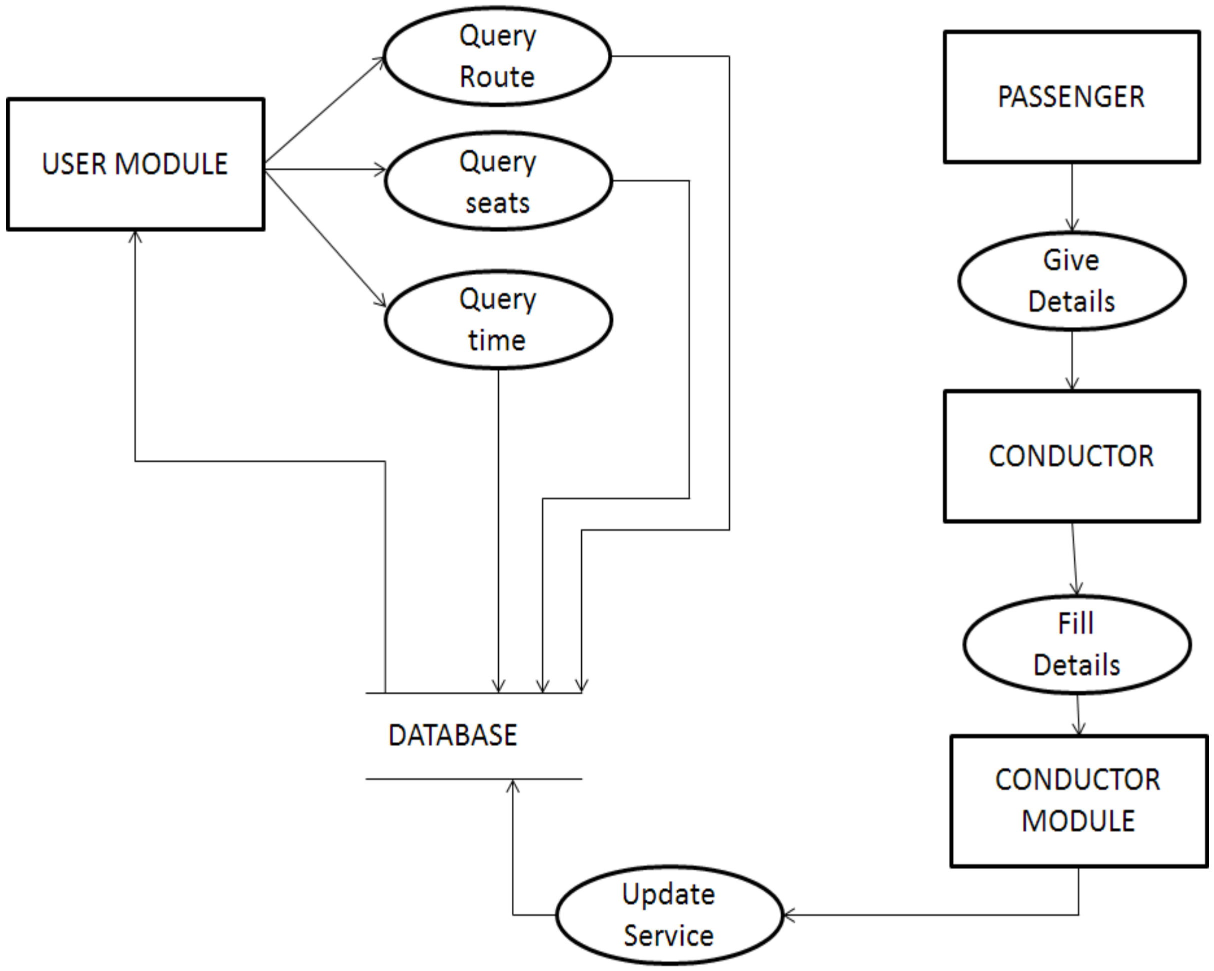


Fig 3.4: DFD LEVEL 1

Fig 3.4 describes the level 1 data flow diagram with more intricate workings of the system. User module includes three queries and conductor takes the details from the passengers and enters in the conductor module. Conductor module has a background update service that keeps the database updated with the recent changes. Conductor plays a very important role in the system since all the response efficiency depends on the data he enters. That would decide the satisfaction of user query responses i.e. whether or not the reality is reflected in the data. User satisfaction is the main need in the whole scenario.

3.4. Activity Diagram

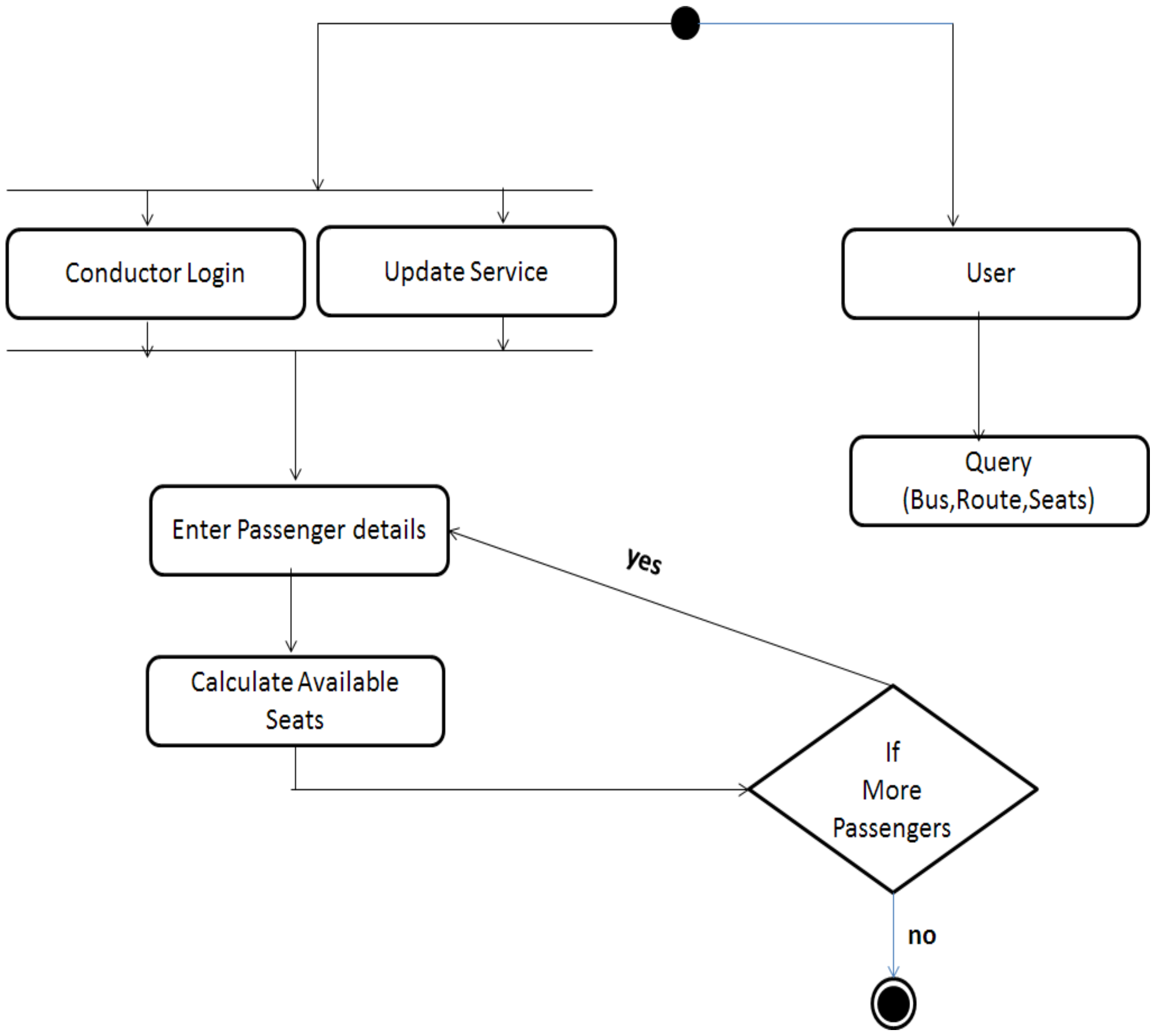


Fig 3.5: Activity diagram

Fig 3.5 is the representation of the activities that occur in the whole system as it works and how do they occur and how the control shifts over different tasks in the system. Firstly the start point goes in two different directions depending on the role you have opted for. For the conductor, entry of passenger details is the task and also the update service that keeps on going to send periodic updates. The other role is of the user which is responsible for asking queries.

3.5 ER Diagrams

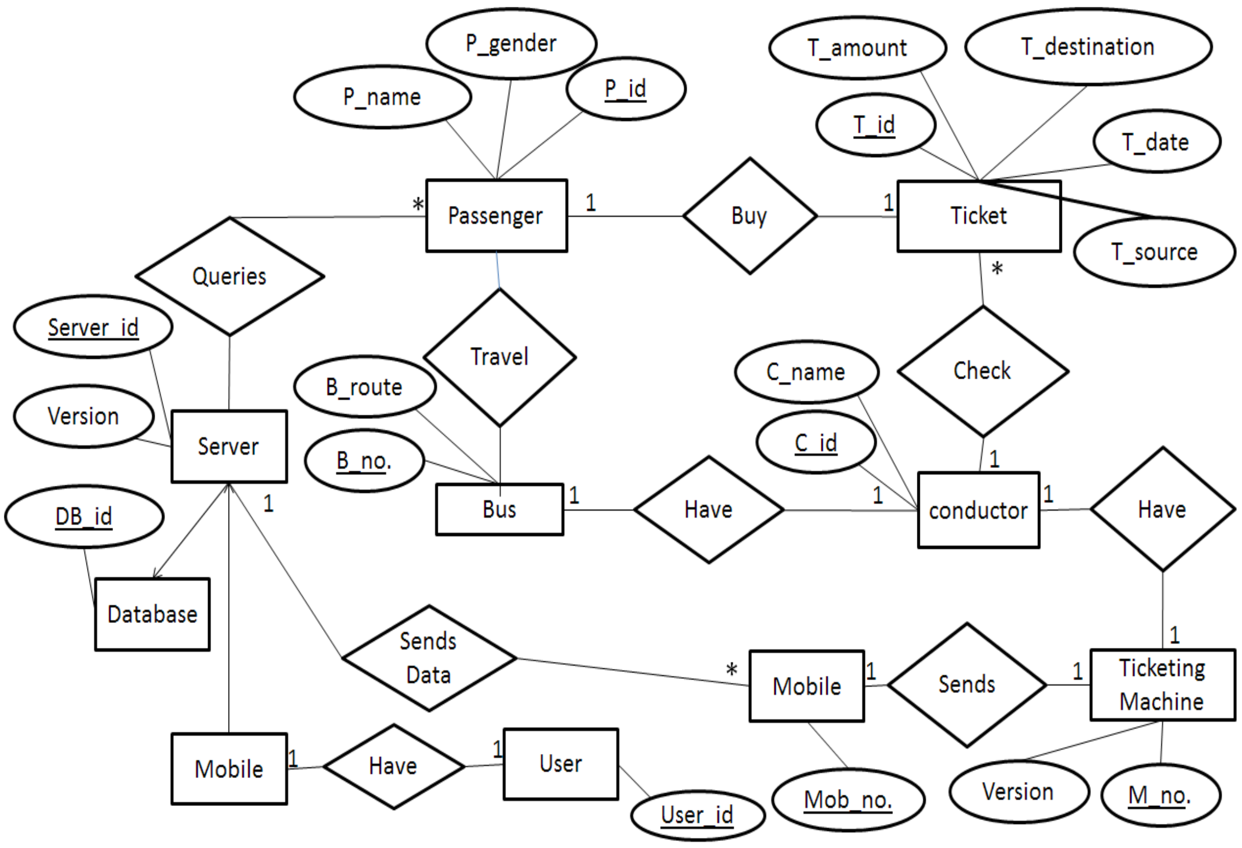


Fig 3.6: ER diagram

Fig 3.6 is an Entity Relationship diagram which is constituted of all the attributes to major actors and the relation between them. The passenger will have a name, a particular id and gender. It has to be noted that the details regarding a passenger can be changed according to the requirements of the real system. The ticket that will be made will also have a unique Id and some other attributes like date, source, destination etc. The relationships between entities like tickets are checked by conductor, passenger buy tickets etc. represents the connection between several activities that will occur dependently.

CHAPTER 4

IMPLEMENTATION AND RESULTS

This chapter focuses on the way project has been implemented. The final results, the snapshots of the application that we have built and the requirements in terms of software and hardware. The importance of this chapter is its emphasis on the outcome of the whole hard work that has been done till now.

4.1 Software and Hardware Requirements

- ADT bundle/Android Studio with Gradle
- Android SDK 24.3.4
- Java SE Development Kit 8
- Android OS Mobile
- Hardware device having GPS and INTERNET

4.2 Assumptions and dependencies

The following must be assumed for ensuring perfect working of the product:

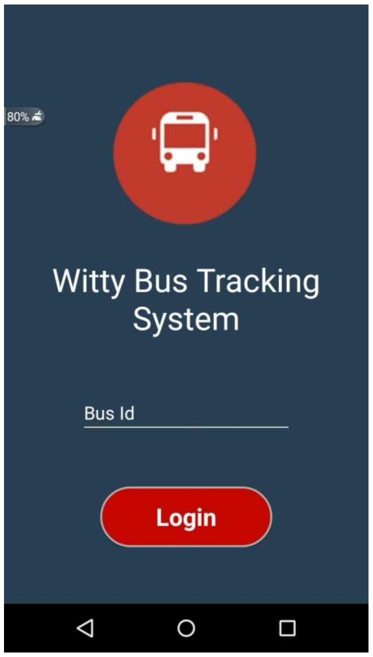
- The ticketing machine supports Bluetooth which is switched on all the time.
- Mobile device embedded in the bus has all time internet access.

4.3 Constraints

- Operating system in mobile device must be android.
- Ticketing machine must support Bluetooth
- Mobile embedded in the bus must have GPS, Bluetooth support and internet access.
- Mobile embedded in the bus must be installed at a secure place where unauthorized access is not allowed.

4.4 Implementation Details

4.4.1 Snapshots Of Interfaces



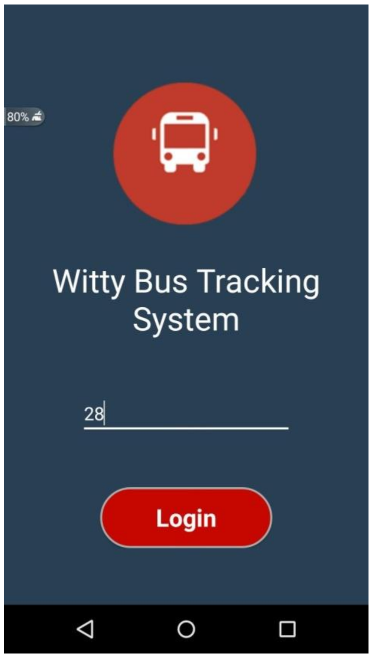
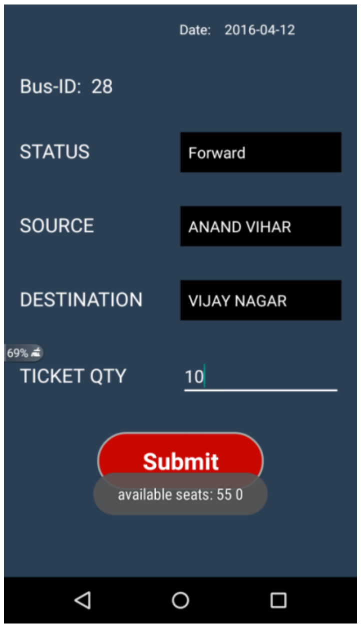


Fig 4.1 **Fig 4.2**

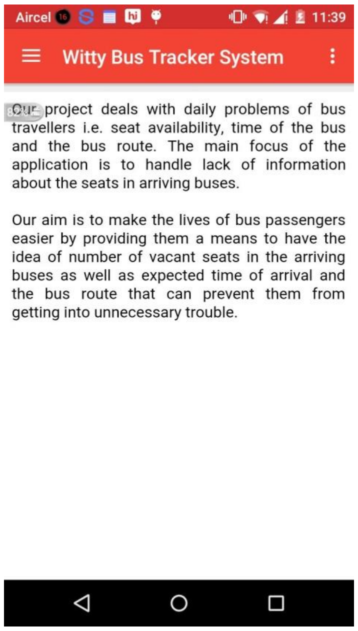
Fig 4.1 and 4.2 show the login screen of conductor module where conductor enters their bus Id which is unique for every existing bus. There is a small logo of the bus and a login button on the screen. Clicking on that button will lead you the activity made for entering passengers details



Date: 2016-04-12

Bus-ID: 28

STATUS	Forward
SOURCE	ANAND VIHAR
DESTINATION	VIJAY NAGAR
TICKET QTY	10



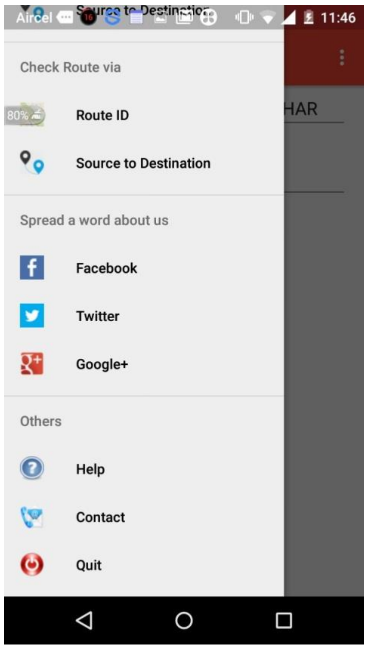
Witty Bus Tracker System

Our project deals with daily problems of bus travellers i.e. seat availability, time of the bus and the bus route. The main focus of the application is to handle lack of information about the seats in arriving buses.

Our aim is to make the lives of bus passengers easier by providing them a means to have the idea of number of vacant seats in the arriving buses as well as expected time of arrival and the bus route that can prevent them from getting into unnecessary trouble.

Fig 4.3 **Fig 4.4**

Fig 4.3 shows the next activity after fig 4.2 where conductor enters user details including source and destination and also the direction in which the bus is going. The backward direction would mean swapping of source and destination. Further the fig 4.4 depicts the user homepage of user module which has first page about the application.



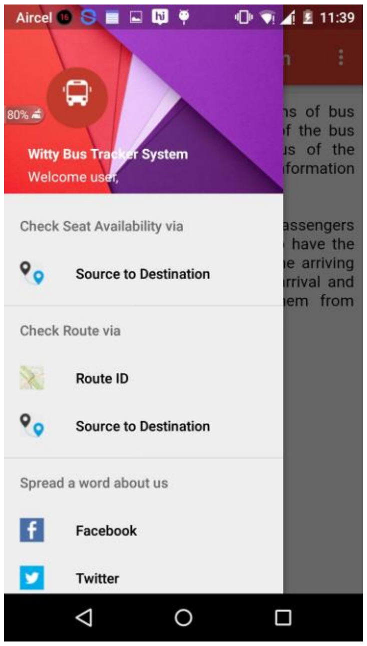
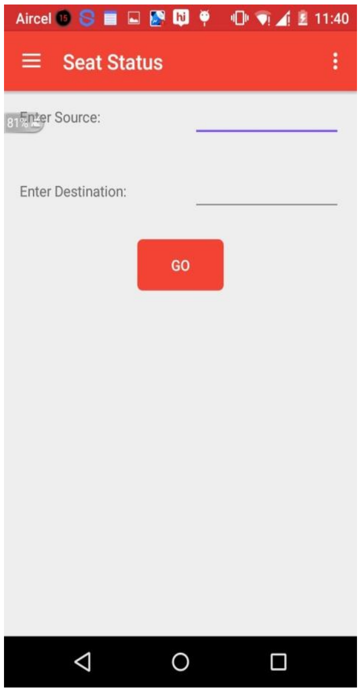


Fig 4.6 **Fig 4.5**

Fig 4.5 and fig 4.6 show the navigation drawer that gives user the choice to select whether he wants to see route or seat availability.



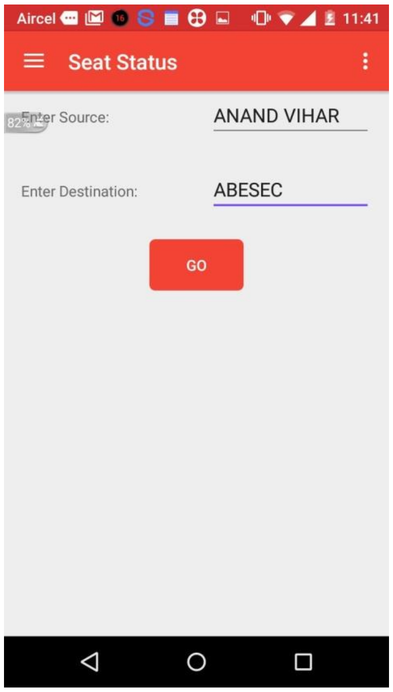
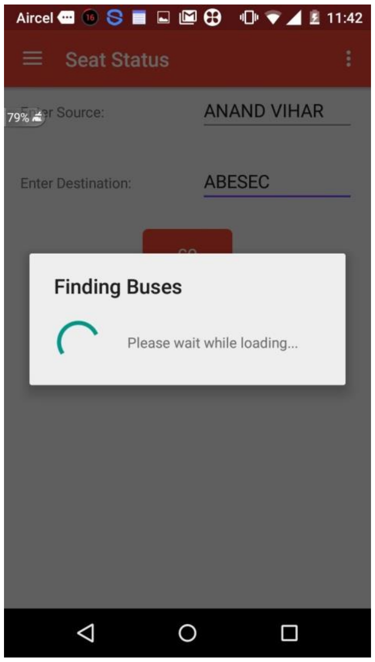


Fig 4.7 **Fig 4.8**

Fig 4.7 and 4.8 show us the fragments if seat status is selected where user will enter Source and Destination. Upon some processing the user will be shown the buses that travel on same or different routes between those entered bus stops. The buses will be shown with their expected time arrival, available seats and extra citizens if any. This has been displayed in fig 4.9 and fig 4.10.


Seat Status

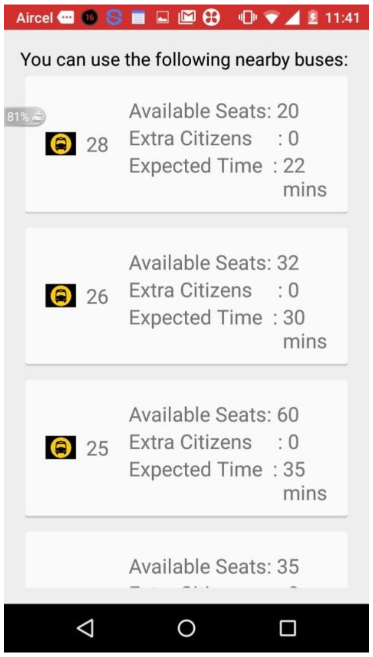
Enter Source: ANAND VIHAR

Enter Destination: ABESEC

Finding Buses

Please wait while loading...

Fig 4.9


You can use the following nearby buses:

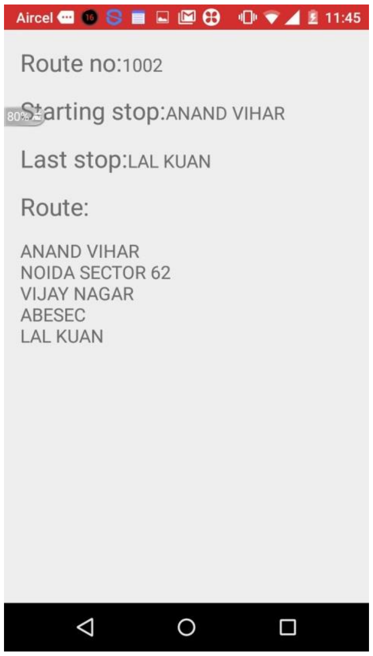
28 Available Seats: 20
Extra Citizens : 0
Expected Time : 22 mins

26 Available Seats: 32
Extra Citizens : 0
Expected Time : 30 mins

25 Available Seats: 60
Extra Citizens : 0
Expected Time : 35 mins

Available Seats: 35

Fig 4.10


Route no:1002

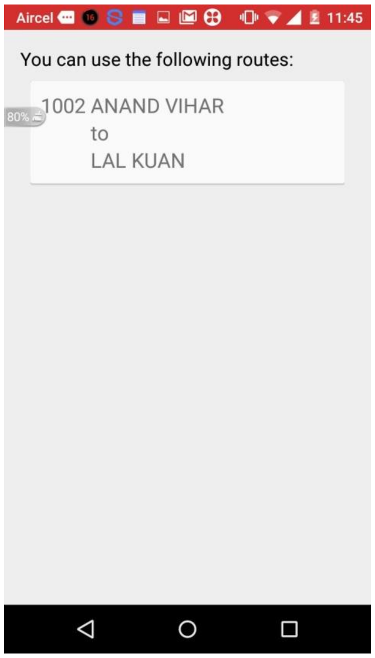
Starting stop:ANAND VIHAR

Last stop:LAL KUAN

Route:

ANAND VIHAR
NOIDA SECTOR 62
VIJAY NAGAR
ABESEC
LAL KUAN

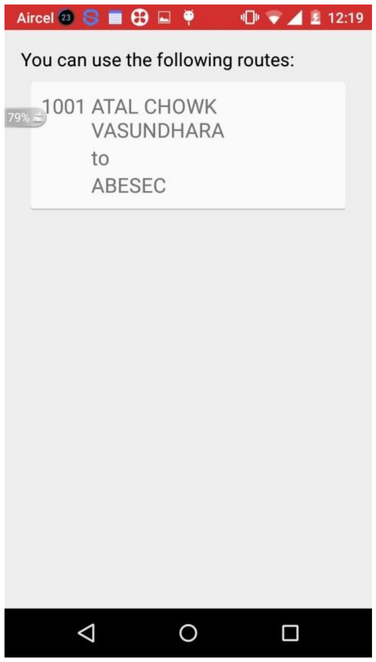
Fig 4.11


You can use the following routes:

1002 ANAND VIHAR
to
LAL KUAN

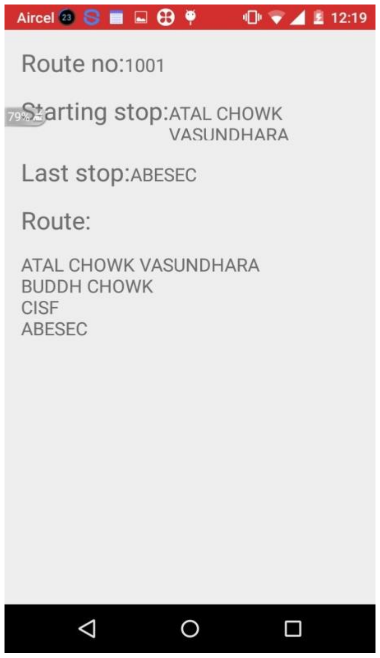
Fig 4.12

User can also see the details if the route Id is known. For example in fig 4.11 the route Id 1002 has been entered by the user and its details are shown in fig 4.12 from the starting point to the destination.


Aircel 12:19

You can use the following routes:

1001 ATAL CHOWK VASUNDHARA to ABESEC


Aircel 12:19

Route no:1001

Starting stop:ATAL CHOWK VASUNDHARA

Last stop:ABESEC

Route:

ATAL CHOWK VASUNDHARA
BUDDH CHOWK
CISF
ABESEC

Fig 4.13 **Fig 4.14**

Similarly fig 4.14 show us the detailed route of route Id 1001 i.e. its starting and stopping places including the intermediate bus stops. The route details can help a passenger to decide a better bus available accordingly.

4.4.2 **Test Cases**

Table 4.1 Test Cases

S.No	**Test steps**	**Test data**	**Expected output**	**Observed output**	**Pass/Fail**
1.	User opens application	Icon is clicked	Home page is opened	Home page is opened	Fail
2.	User wants to enter queries	Open navigation drawer	Drawer opens	Drawer slides from left with options	Fail
3.	User wants to know route	Click on option with source and destination	Navigated to new fragment with corresponding entries	New fragment successfully opens	Fail
		Click on option with route Id	Navigated to new fragment with corresponding entry	New fragment successfully opens	Fail
4.	User wants to know vacant seats	Click on option with seat availability	Fragment entering source and destination	New fragment successfully opens	Fail
5.	User enters source and destination(Route)	ANAND VIHAR LAL KUAN	The whole route from source to destination	ANAND VIHAR NOIDA SECTOR 62 VIJAY NAGAR ABESEC LAL KUAN	Fail

6.	User enters source and destination(seat availability)	ANAND VIHAR LAL KUAN	All the available buses with number of vacant seats, extra passengers, expected time of arrival	Details of buses are displayed	Fail
7.	User enters route ID(Route)	1001	Route opens	ATAL CHOWK VASUNDHARA BUDDH CHOWK CISF ABESEC	Fail
				ANAND VIHAR LAL KUAN	Fail
		1002	Route opens		

4.4.3 Results

The project “witty bus tracker system” has successfully resulted in exhibiting proper outputs as a prototype. With the required conditions the android application produces acceptable outputs.

In Table 4.2 the criteria’s like effective time, route, vacant seats, in cities that it operates etc. we have compared our application to the existing applications. Effective time implies the time that a bus will require to reach a stop in its current running state including traffic not the time at which it is supposed to reach. Route details are whether the app gives the complete details of all the stops between source and destination. Vacant seats are the number of currently available seats in the bus. Also we have seen if the existing app is for only one state or more than 1.

Table no. 4.2 Comparisons

S.No	**App Name/Features**	**Effective Time**	**Route**	**Vacant Seats**	**In more than one state**
1.	BMTC	No	Yes	No	No
2.	MTC Bus Route	No	Yes	No	No
3.	Kolkata Bus Info	No	Yes	No	No
4.	City Bus	No	Yes	No	Yes
5.	Witty Bus Tracker System	Yes	Yes	Yes	NA

CHAPTER 5

CONCLUSION

This chapter concludes our efforts. Specific features of the application that makes it different than the existing technologies have been explained. The future aspects of our application have also been discussed.

5.1Performance

Some of the specific features that our application holds

- Easy UI- The interface of application has been designed as it would be easy to use for all categories including all ages and different literacy levels.
- Tells vacant seats- Telling a passenger about the current approximate seats is a special feature as it has not been implemented till date.
- Tells route- telling passenger about the route of the bus that is all the stops that the bus is supposed to make.

5.2Comparison with existing State-of-the-Art Technologies

A technology that helps people through making their travelling experience better is hard to compare. Also the fact that apps that can tell you timings and route of buses have been made but no such app exists that can tell the number of vacant seats so it’s safe to say that this exact technology does not exists till date. The full comparison of our application with already running apps is mentioned in Table 4.2 including the comparison of their features.

5.3Future Directions

Our project is a prototype designed for future development regarding bus services availed by people in different parts of the country. Since such a service does not exist till date for regular buses our prototype could act as a base for such kind of development.

REFERENCES

1. Shi-yi Xie Coll. of Inf. Technol., Zhanjiang Ocean Univ., China Sheng Gao, Bing Xu "*Study of an optimum scheduling algorithm about buses in city intelligent transport systems*." IEEE Communications Machine Learning and Cybernetics, 2004.

2. Eken, S. ; Dept. of Comput. Eng., Kocaeli Univ., Kocaeli, Turkey ; Sayar, A. "*A smart bus tracking system based on location-aware services and QR codes"*. IEEE Communications Society Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014."

3. Kostakos, V. ; Madeira Interactive Technol. Inst., Univ. of Madeira, Funchal, Portugal ; Camacho, T. ; Mantero, C. "*Wireless detection of end-to-end passenger trips on public transport buses*". IEEE Communications Society Intelligent Transportation Systems (ITSC), 2010.

4. "Adding the App Bar" January 15, 2015, http://developer.android.com/training/appbar/index.html
5. "Creating Swipe Views With tabs" January 15, 2015, http://developer.android.com/training/implementing-navigation/lateral.html
6. Ravi Tamada ,"Android material design working with tabs" September 13, 2015, http://www.androidhive.info/2015/09/android-material-design-working-with-tabs/
7. Mohit Gupt,"Android tabs example-with fragments and view pager" Jun 7, 2015 http://www.truiton.com/2015/06/android-tabs-example-fragments-viewpager/
8. Ravi Tamada,"Android GPS, location manager tutorial" Jan 1, 2012 http://www.androidhive.info/2012/07/android-gps-location-manager-tutorial/
9. "Location Strategies "January 18, 2015, http://developer.android.com/guide/topics/location/strategies.html